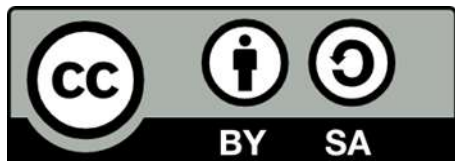


Web Server

Silvio Peroni
speroni@cs.unibo.it



<http://creativecommons.org/licenses/by-sa/2.5/it/>

Sommario

- Web Server
- Apache HTTP Server
- Configurazione, log e autenticazione
- Script CGI e gestione degli utenti
- URL Rewriting
- SSL e Apache

Cos'è un Web server

- Un **Web server** è un processo che si occupa di ricevere richieste HTTP da parte di un client (es: un browser) e di fornirne una risposta HTTP
- La risposta del offerta può prevedere dei dati opzionali. Es: pagina Web
- I due più famosi Web server in circolazione sono (dati aggiornati a gennaio 2009):
 - ✦ Apache HTTP Server (52.05% di siti ospitati)
 - ✦ Microsoft IIS (32.9 % di siti ospitati)

Apache HTTP Server

- **Apache HTTP Server** è il Web server HTTP open-source della Apache Software Foundation
- Dal 1996, è il Web server più diffuso su Internet
- L'ultima versione è la 2.2.11, rilasciata il 14 dicembre 2008
- È un server **modulare**: le funzionalità sono rese disponibili tramite moduli attivabili o disattivabili a piacimento:
 - ✦ staticamente, in fase di compilazione
 - ✦ dinamicamente, modificando il file "httpd.conf" sfruttando il supporto *Dynamic Shared Object* con la direttiva "LoadModule")

Apache su Unix (Linux & Mac)

- In Unix, Apache è un programma “httpd” eseguito in background come demone e si occupa di gestire le richieste HTTP
- “httpd” può essere eseguito lanciando lo script “apachectl” che si occupa di configurare anche alcune variabili di ambiente
- Lo script precedente permette il controllo del ciclo di vita del web server mediante l’uso delle seguenti quattro operazioni:
 - ✦ **start**: avvia il server
 - ✦ **stop**: blocca l’esecuzione del server
 - ✦ **restart**: ri-avvia il server
 - ✦ **graceful restart**: termina tutti i processi figlio del server e poi lo riavvia

Apache su Windows

- Pacchetti possibili in “.zip” (sorgenti) e in “.msi”
- Due possibili modalità per lanciare il server
 - ✦ aprire la shell (DOS) e lanciare “httpd.exe” seguito eventualmente da opportuni parametri
 - ✦ tramite l'interfaccia grafica di Windows
- Esistono svariati pacchetti (es: EasyPHP) che includono sia Apache sia MySQL già preconfigurati

Configurare Apache

- Il file usato per specificare la configurazione principale di Apache è “httpd.conf”
- Le direttive specificabili in “httpd.conf” si dividono in tre gruppi fondamentali:
 - ✦ direttive che controllano il server come processo
 - ✦ direttive che definiscono i parametri per il server principale (home directory, index, ecc.)
 - ✦ direttive che configurano gli eventuali server virtuali
- Tutte le direttive specificabili sono definite in <http://httpd.apache.org/docs/2.2/mod/directives.html>
- Le linee del file di configurazione che iniziano per “#” sono commenti

Altri file di configurazione

- La gestione dei tipi MIME viene gestita nel file “mime.types”
- La gestione per la configurazione di ogni sito o cartella è gestita mediante file (opzionali) chiamati “.htaccess”
 - ✦ le direttive di un file “.htaccess” sono applicate solo alle sotto cartelle in cui è presente
 - ✦ le modifiche al file vengono caricate dinamicamente
 - ✦ **può essere configurato anche dagli utenti**, non solo dagli amministratori
 - ✦ la direttiva “AllowOverride” permette agli amministratori di controllare il “potere” di “.htaccess”

I file di log

- I file di log permettono di controllare il flusso di richieste/risposte HTTP
- “/logs/access.log” registra gli accessi avvenuti con successo
- “/logs/error.log” registra, descrivendoli brevemente, gli errori
- “/logs/rewrite.log” registra le modifiche agli URL delle richieste. Non esiste di default ma deve essere indicato esplicitamente; inoltre può essere attivato esclusivamente dagli amministratori modificando il file “httpd.conf”

Autenticazione

- Per permettere l'autenticazione sul Web server è necessario creare o un file contenente le password (su Unix, mediante il comando "htpasswd") o un database per la loro gestione (mediante l'attivazione del modulo "mod_auth_dbm")

AuthType Basic

AuthName "Restricted Area"

AuthUserFile /usr/local/apache/passwd/password

require user AngeloDiIorio, group GruppoTW (o valid-user)

- L'abilitazione all'accesso può essere anche determinata dall'*host-address* della macchina richiedente

Allow from 205.252.46.165

Deny from 205.252.46.165

Script CGI in Apache

- La *Common Gateway Interface* (CGI) fornisce al server un meccanismo per interagire con programmi esterni nella creazione dei contenuti attraverso l'uso di script
- In Apache due possibili metodi per specificare l'utilizzo di CGI:
 - ✦ definire – mediante la direttiva “ScriptAlias” – una cartella contenente gli script i quali devono avere i diritti di esecuzione
 - ✦ permettere l'esecuzione degli script in una cartella specifica

```
<Directory /usr/local/apache/htdocs/somedir>  
    Options + ExecCGI  
</Directory>
```

Gestione degli utenti

- C'è la possibilità di definire, per ogni utente del sistema, una directory nel quale rendere disponibili sul Web i propri documenti. Per fare ciò si usa la direttiva “UserDir” specificando il path desiderato
- È possibile abilitare gli utenti all'uso di script CGI personali

```
<Directory /home/somepath/cgi-bin>  
    Options + ExeCGI  
    SetHandler cgi-script  
</Directory>
```

URL Rewriting

- Apache fornisce opportuni moduli per manipolare gli URL in modo da consentirne l'elaborazione interna o la redirectione
- La manipolazione dell'URL può essere comoda per facilitarne la lettura e l'interpretazione da parte degli utenti
- **La maggior parte delle direttive può essere specificata nel file “.htaccess”**
- Quale dei due preferite?
 - ✦ `http://www.sito.it/index.php?id=20081204-3&t=post&u=donald`
 - ✦ `http://www.sito.it/donald/post/2008/12/04/3`

mod_alias

- Questo modulo, attivo di default in Apache, permette di creare sia *alias* sia *redirect* degli URL che arrivano come richiesta ad Apache
- Le direttive “Alias” e “ScriptAlias” permettono di mappare URL su percorsi del file system. Mentre con la prima si prendono come riferimento URL generici, la seconda permette di specificare la cartella di arrivo come “CGI directory”. Definibili soltanto in “httpd.conf”
- La direttiva “Redirect” è usata per dire al client di effettuare una nuova richiesta ad un URL differente. **Definibili anche in “.htaccess”**

Ordine di processamento delle direttive

- Le “Redirect” vengono processate prima di ogni “Alias”. Conseguentemente, la presenza di “Redirect” in un dato contesto annulla completamente l’esecuzione dei successivi “Alias”
- “Alias” e “Redirect” vengono applicate secondo l’ordine in cui appaiono sul file di configurazione. Per questa ragione, quando ci sono due o più direttive agenti sullo stesso sotto-percorso, bisogna elencare prima il percorso più specifico in modo che tutte le direttive seguenti abbiano effetto

```
Alias /pippo/pluto /cani  
Alias /pippo /personaggio
```

vengono eseguite entrambe

```
Alias /pippo /personaggio  
Alias /pippo/pluto /cani
```

viene eseguita soltanto la prima

Alias

- Permettono la definizione di alias in modo da mappare un particolare URL in un particolare percorso del file system
- Sintassi: `Alias URL-path file-path|directory-path`
- Esempio: `Alias /img/ /pub/img/png/`

ScriptAlias

- Permettono la definizione di alias in modo da mappare un particolare URL in una particolare percorso riferito esclusivamente a script CGI
- Sintassi: `ScriptAlias URL-path file-path|directory-path`
- Esempio: `ScriptAlias /script/ /web/cgi-bin/`

Redirect

- Permettono di mappare un vecchio URL in uno nuovo chiedendo al client di ripetere la richiesta alla nuova locazione
- Sintassi: `Redirect [status] URL-path URL`
- Esempio: `Redirect /home http://www.new.it/home`
- Valori possibili per lo status:
 - ✦ *permanent*, ritorna uno stato di redirectione permanente (301)
 - ✦ *temp* (default), ritorna uno stato di redirectione temporanea (302)
 - ✦ *seeother*, indica che la risorsa è stata sostituita (303)
 - ✦ *gone*, indica che la risorsa è stata permanentemente rimossa (304)

Espressioni regolari

- Facendo seguire alle tre direttive precedenti la parola “Match” – “AliasMatch”, “ScriptAliasMatch” e “RedirectMatch” – si può far uso di espressioni regolari per l’identificazione degli URL
- La sintassi rimane la stessa delle precedenti soltanto che, invece di specificare un *URL-path* si specifica un’espressione regolare
- Posso inoltre usare, nell’istruzione terminale di arrivo dell’alias o del redirect, istruzioni – \$1, \$2, \$3, ecc. – che mi permettono di utilizzare parte dell’URL di partenza (ogni stringa specificata tra parentesi tonde, numerate secondo l’ordine in cui compaiono)

```
AliasMatch ^/img/(.*) /img/png/$1
```

```
AliasMatch ^index.php?id=(.*)&t=(.*)&u=(.*) /$3/$2/$1
```

Caratteri speciali

Character	Meaning	Example
.	Matches any single character	<code>c.t</code> will match <code>cat</code> , <code>cot</code> , <code>cut</code> , etc.
+	Repeats the previous match one or more times	<code>a+</code> matches <code>a</code> , <code>aa</code> , <code>aaa</code> , etc
*	Repeats the previous match zero or more times.	<code>a*</code> matches all the same things <code>a+</code> matches, but will also match an empty string.
?	Makes the match optional.	<code>colou?r</code> will match <code>color</code> and <code>colour</code> .
^	Called an anchor, matches the beginning of the string	<code>^a</code> matches a string that begins with <code>a</code>
\$	The other anchor, this matches the end of the string.	<code>a\$</code> matches a string that ends with <code>a</code> .
()	Groups several characters into a single unit, and captures a match for use in a backreference.	<code>(ab)+</code> matches <code>ababab</code> - that is, the <code>+</code> applies to the group. For more on backreferences see below .
[]	A character class - matches one of the characters	<code>c[ua]t</code> matches <code>cut</code> , <code>cot</code> or <code>cat</code> .
[^]	Negative character class - matches any character not specified	<code>c[^/]t</code> matches <code>cat</code> or <code>c-t</code> but not <code>c/t</code>

mod_rewrite

- Questo modulo, molto più potente di “mod_alias”, deve essere esplicitamente attivato in Apache
- Al contrario di suo fratello minore, tutte le direttive di “mod_rewrite” possono essere messe sia in “httpd.conf” sia in “.htaccess”
- Possibilità di specificare *flag* in modo da creare differenti relazioni tra la sequenza
- Possibilità di specificare condizioni particolari in cui deve avvenire o non deve avvenire la riscrittura

Attivazione del motore

- Se si vuol usare il motore di riscrittura è necessario attivarlo esplicitamente nei file di configurazione desiderati mediante la direttiva “RewriteEngine” specificata a “On”
- L'ordine di esecuzione delle regole avviene a cascata, tenendo in forte considerazione eventuali *flag* specificate per ognuna

```
RewriteEngine On
```

```
RewriteRule ^index.php?id=(.*) /data/$1
```

RewriteRule

- Permettono la definizione di regole di riscrittura in modo da trasformare una stringa, dell'URL richiesto dal client, in una nuova
- Sintassi: `RewriteRule regex substitution [flag]`
- Alcune delle flag specificabili sono le seguenti:
 - ✦ `[C]`, collega la regola corrente con la seguente. Nel caso la regola corrente venga saltata a seguito di una valutazione, allora anche la seguente verrà automaticamente saltata
 - ✦ `[F]`, indica che l'URL definito da questa regola è proibito (403)
 - ✦ `[L]`, ferma il processo di riscrittura dopo l'esecuzione della regola corrente
 - ✦ `[N]`, fa ripartire il processo di riscrittura da capo ma con l'URL modificata a seguito delle applicazioni delle regole precedenti
 - ✦ `[P]`, la sostituzione viene effettuata e passata come richiesta di proxy verso il modulo gestore del proxy (proxy_mod)
 - ✦ `[S=n]`, salta le n regole successive a questa

RewriteCond

- Serve per specificare una condizione prima di applicare la regola di riscrittura da cui è seguita
- Sintassi: `RewriteCond TestString CondRegex [flag]`
- *TestString* può contenere riferimenti
 - ✦ a regole di riscrittura precedenti (`$N`, `N` da 0 a 9)
 - ✦ a condizioni precedenti (`%N`, `N` da 1 a 9)
 - ✦ a variabili del server (`%{NAME_OF_VARIABLE}`, come `HTTP_USER_AGENT`, `HTTP_HOST`, ecc)
 - ✦ a direttive “`RewriteMap`” (definibili solo su “`httpd.conf`”)
- *CondRegex* può iniziare con confrontatori numerici o lessicografici (“`<`”, “`>`”, “`=`”) o operatori di test (“`-d`”, “`-f`”, ecc.)
- I *flag* possono essere usati per concatenare più condizioni (es: `[OR]`)

Esempio

RewriteEngine On

RewriteCond %{HTTP_USER_AGENT} ^Mozilla.*

RewriteRule ^(.*)\$ /mozilla/\$1

RewriteRule ^/(.*)\.html\$ /index.php?p=(\$1)

RewriteRule ^/~([a-z])/([a-z]+)/(.*) /home/\$2/\$1/.www\$3

RewriteRule ^/~(.+) http://newserver/~\$1

RewriteCond %{TIME_HOUR}%{TIME_MIN} >0700

RewriteRule ^home\.htm\$ home_day.html

SSL

- La Apache Software Foundation non include nel progetto Apache Server il modulo per il supporto di SSL
- Esistono due progetti open source che si propongono di garantire il supporto di SSL, entrambi basati su **OpenSSL: mod_ssl**, un modulo installabile nel server; **Apache-SSL**, un server con il supporto SSL integrato
- Esistono altre implementazioni commerciali di moduli SSL per Apache: "Covalent Raven SSL Module For Apache" o la distribuzione IBM di Apache "IBM Http Server"

Direttive per SSL

- **SSLSessionCache**: imposta il tipo di memorizzazione della cache
- **SSLEngine**: abilita/disabilita le connessioni SSL
- **SSLProtocol**: controlla la versione del protocollo da usare. Es. SSLProtocol All -SSLv2
- **SSLCertificateFile** e **SSLCertificateKeyFile**: contengono il path ai file certificato e chiave del server
- **SSLRequireSSL**: usata nelle sezioni "<directory>" richiede connessioni sicure per accedere ad alcune aree del server
- **SSLCipherSuite** permette di configurare gli algoritmi crittografici usati nella fase iniziale di scambio della chiave

Bibliografia

- Web Server, articolo Wikipedia - http://en.wikipedia.org/wiki/Web_server
- Apache HTTP Server Documentation - <http://httpd.apache.org/docs/2.2/>
- File “httpd.conf” di esempio - <http://www.ccl.net/cca/software/UNIX/apache/solaris-t3.2b5/files/http.conf-init.shtml>
- .htaccess, articolo Wikipedia - <http://en.wikipedia.org/wiki/.htaccess>

Bibliografia

- mod_alias - http://httpd.apache.org/docs/2.2/mod/mod_alias.html
- Espressioni regolari - <http://www.pcre.org/>
- mod_rewrite - http://httpd.apache.org/docs/2.2/mod/mod_rewrite.html
- Documentazione dettagliata per mod_rewrite - <http://httpd.apache.org/docs/2.2/rewrite/>